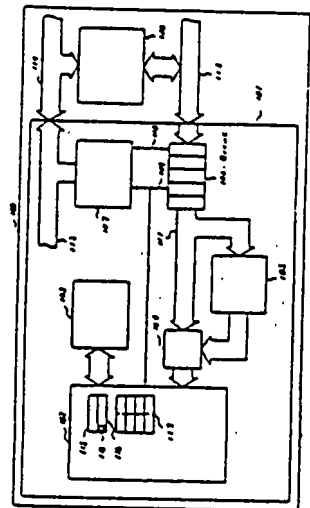


(54) INSTRUCTION PROCESSING SYSTEM
(11) 62-25334 (A) (43) 3.2.1987 (19) JP
(21) Appl. No. 60-165177 (22) 25.7.1985
(71) NEC CORP (72) YUKARI MISAWA(1)
(51) Int. Cl. G06F9/44, G06F9/30, G06F15/06

PURPOSE: To perform in parallel the storage of the 1st and the 2nd instruction code groups in a queue storage means, conversion from the 2nd instruction code to the 1st instruction code, and the execution of the 1st instruction code.

CONSTITUTION: An instruction code on an internal data bus 111 is the instruction code of a host device in native mode and an execution unit 102 is in the native mode, so an instruction code selector 104 selects the instruction code on the internal data bus 111 and the execution unit 102 decodes the instruction code to perform processing. In emulation mode, on the other hand, the instruction code on the internal data bus 111 is the instruction code of a low priority device and this instruction code is sent to a code conversion memory 105 and converted into the instruction code of the host device. The instruction code selector 104 selects the instruction code in the code conversion memory 105 and outputs the instruction code which is converted into a host device instruction to the execution unit 102.



103: input/output device, 107: fetch unit, 108: memory

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A)

昭62-25334

⑬ Int. Cl.⁴

識別記号

庁内整理番号

⑭ 公開 昭和62年(1987)2月3日

G 06 F 9/44
9/30
15/06

A-8120-5B
A-7361-5B
7343-5B

審査請求 未請求 発明の数 1 (全7頁)

⑮ 発明の名称 命令処理方式

⑯ 特 願 昭60-165177

⑰ 出 願 昭60(1985)7月25日

⑱ 発 明 者 三 沢 ゆ か り 東京都港区芝5丁目33番1号 日本電気株式会社内
⑲ 発 明 者 香 取 重 遠 東京都港区芝5丁目33番1号 日本電気株式会社内
⑳ 出 願 人 日本電気株式会社 東京都港区芝5丁目33番1号
㉑ 代 理 人 弁理士 内 原 晋

明 細 書

1. 発明の名称

命令処理方式

2. 特許請求の範囲

プログラム記憶手段内に格納される第1の命令コード群を解釈する解釈回路の制御により、各種のデータ処理を実行する演算処理装置を単一半導体基板上に集積した半導体集積回路において、複数の前記第1の命令コード群と第2の命令コード群を蓄える記憶手段と、前記プログラム記憶手段内の前記第1の命令コード群と前記第2の命令コード群を前記記憶手段へ送出する命令読み出し手段と、前記第1の命令コード群を実行する実行手段と、前記第2の命令コード群から前記第1の命令コード群を生成する命令コード変換記憶手段とを含み、前記命令読み出し手段による前記プログラム記憶手段内の前記第1の命令コード群と前記第2の命令コード群の前記記憶手段への格納処理

と、前記命令コード変換記憶手段による前記第2の命令コードから前記第1の命令コードへの変換処理と、前記実行手段による前記第1の命令コードの実行処理とを並列に実行することを特徴とする命令処理方式。

3. 発明の詳細な説明

(産業上の利用分野)

本発明は、命令コードを解釈する解釈回路の制御により各種のデータ処理を実行する演算処理装置を単一半導体基板上に集積した情報処理装置の命令処理方式に関する。

(従来の技術)

最近、パーソナルコンピュータ、オフィスコンピュータ等の情報処理装置が、企業内ではもちろん、一般家庭でも多く利用されている。これら情報処理装置は、利用範囲が広く、それに伴うソフトウェアも増え続け、今では膨大な数となっている。

ソフトウェアの増大と平行して、情報処理装置

も利用者の要求とともに、高性能化、高速処理化が進んでいる。

しかし、次々と新しい機種が製品化されているため、最新機種として製品化された情報処理装置もすぐに旧式化しているというのが現状である。

従来の機種(以下、下位機種と記す)には、多くのソフトウェアが用意されているが下位機種の命令コードは、最新の機種(以下、上位機種と記す)の命令コードとは全く異なる。最新の上位機種のソフトウェアと従来の下位機種のソフトウェアとの互換性が問題になっている。

第8図は、従来の情報処理装置の構成図である。CPU801は上位機種の命令セットを実行するマイコンである。メモリ802には、実行されるプログラムや処理データが格納されており、バスでCPU801や入出力装置802と結合されている。

上記の様な構成の装置で下位機種のソフトウェアを実行してもCPU801は上位機種の命令コードとして解釈し正しい処理を行うことは不可

能である。

また、この膨大な数の従来の下位機種用のソフトウェアが上位機種に使用できないという理由で無視することも不可能である。そこで、今まで上位機種上で下位機種用のソフトウェアを使用する方法として、第1に下位機種用ソフトウェアのソースプログラムを上位機種用に書き直して上位機種上で用いる方法、第2に上位機種用の命令で全く新しいソフトウェアを作るという方法があった。

しかし、第1の方法には、現在使われているすべての下位機種用のソフトウェアを上位機種用に書き直すために大変な時間を費してしまうという問題がある。また、第2の方法には、従来のソフトウェアと同じだけのソフトウェアを新たに作成するために今までと同じ、または、それ以上の工数と費用がかかってしまうという問題がある。つまり、上記の2つの方法は、下位機種用のソフトウェアが膨大な数となってしまった今では用いることが不可能である。そこで従来は、第9図の様な構成の情報装置900が考えられた。

次に第9図を参照しながら情報処理装置900の構成と動作を説明する。

第9図は、上位機種の命令セットを実行するマイコン901と下位機種の命令セットを実行するマイコン902の両方を搭載させた情報処理装置900のブロック図である。第9図において、上位機種の命令セットを実行するマイコン901と下位機種の命令セットを実行するマイコン902は、エミュレーション制御装置903に接続されている。上位機種用のソフトウェアを実行させる時は、エミュレーション制御装置903の制御でアドレスバス905、データバス906が上位機種の命令セットを実行するマイコン901に接続され、メモリ904の中の上位機種用のプログラムを実行する。下位機種用のソフトウェアを実行させるときは、エミュレーション制御装置903の制御でアドレスバス905、データバス906が下位機種の命令セットを実行するマイコン902に接続され、メモリ904の中の下位機種用のプログラムを実行する。この様に、下位機種用、

上位機種用両方のソフトウェアを実行できる。

(発明が解決しようとする問題点)

上述した従来の情報処理装置は、上位機種の命令コード実行用と、下位機種の命令コード実行用の2つのマイコンを搭載しており、増設型装置のシステム規模が大きくなってしまいうという欠点がある。それに伴い、価格が高価になってしまいうという欠点もある。また、マイコンを2つ搭載するということは、下位機種の命令コード実行用のマイコンを使っているときは、もう1つの上位機種の命令コード実行用のマイコンは使われておらず、上位機種の命令コード実行用のマイコンを使っているときは、もう1つの下位機種の命令コード実行用のマイコンは使われていないというように、常に片方のマイコンしか使われていないため、ハードウェアが無駄になってしまうという欠点がある。

(問題点を解決するための手段)

本発明に基づく情報処理装置は、プログラム記憶手段内に格納される第1の命令コード部を解釈

する解読回路の制御により、各種のデータ処理を実行する演算処理装置を単一半導体基板上に集積した半導体集積回路において複数の第1の命令コード群と第2の命令コード群を蓄える洋を行列記憶手段と、プログラム記憶手段内の第1の命令コード群と第2の命令コード群を洋ち行列記憶手段へ送出する命令読み出し手段と、第1の命令コード群を実行する実行手段と、第2の命令コード群から第1の命令コード群を生成する命令コード変換記憶手段を有し、命令読み出し手段によるプログラム記憶手段内の第1の命令コード群と第2の命令コード群の洋ち行列記憶手段への格納処理と命令コード変換記憶手段による第2の命令コードから第1の命令コードへの変換処理と実行手段による第1の命令コードの実行処理を並列に実行することを特徴とする。

(実施例)

次に、本発明について図面を参照して説明する。

第1図は、本発明に関する情報処理装置の構成図である。情報処理装置100には2つのモード

があり、1つは上位機種用の命令として実行するモード(以下ネイティブモードと記す)と他方は、下位機種用の命令として処理するモード(以下エミュレーションモードと記す)である。第1図の情報処理装置100のマイクロコンピュータ101は、下位機種の命令セットを実行する機能(以下エミュレーション機能と記す)と本来の命令セットを実行する機能の両方を備えている。マイクロコンピュータ101は実行ユニット102、入出力装置103、命令コードセレクト104、コード変換メモリ105、QUEUE106、フェッチユニット107から構成される。QUEUE106は、複数の命令を蓄えておくことができ、読み出されるときは、書き込まれた順に1命令ずつ内部データバス111へ送出する。QUEUE106からは、QUEUE106の状態信号であるQRDY信号109とQFULL信号110が送出されている。QRDY信号109は、実行ユニット102とフェッチユニット107に送出されており、QUEUE106の中に処理する命令コード

があるかないかを示している。処理する命令コードがある場合は、アクティブレベル"1"となる。QFULL信号110は、フェッチユニット107に送出されており、QUEUE106の中に、まだ、命令コードを書き込める余裕があるかないかを示している。QUEUE106がいっぱいで、命令コードを書き込むことができない場合、アクティブレベル"1"となる。QUEUE106を用いない場合、情報処理装置は第4図の様に動作し、実行ユニット102は、命令コードの実行を終えるまでメモリ108から次の命令コードを読み出すことができないため、情報処理装置内では、常に、フェッチまたは実行の一方の処理しか行うことができない。ところがQUEUE106があると、実行ユニット102が命令コードの実行をしていても、フェッチユニット107の指示でメモリ108からのフェッチが行われ、QUEUE106に蓄えられる。QRDY信号109がアクティブレベル"1"であればQUEUE106からは命令コードが次々と送出されるため、

前の命令コードが実行ユニット102に取り込まれると、次の命令はすでにQUEUEから内部データバス111に送出されている。そのため、第5図の様に、フェッチと実行が平行して行われ、第4図のQUEUEがない場合に比べて処理時間が短縮される。

フェッチユニット107は、QUEUE106からのQRDY信号109およびQFULL110を受けて、メモリ108からQUEUE106への命令の書き込みを制御する。QFULL信号110がインアクティブレベル"0"で、QRDY信号109がアクティブレベル"1"の場合、QUEUE106にはまだ命令を書き込む余裕があり、QFULL信号110がインアクティブレベル"0"で、QRDY信号109もインアクティブレベル"0"の場合、QUEUE106には処理する命令が入っていないため、フェッチユニット107からは、アドレス情報が内部アドレス113上へ送出され、外部アドレスバス114を介してメモリ108に受け取られる。するとメモ

リ108から命令コードが送出され、外部データバス112を経てQUEUE 106に書き込まれ、書えらる。QUEUE 106がいっぱいになると、QFULSH 信号110がアクティブレベル“1”となり、フェッチユニット107は、アドレス情報の送出をしなくなる。QFULSH 信号110がインアクティブレベル“0”になると再びメモリ108からQUEUE 106へ命令コードの書き込みが行われる。

コード変換メモリ105には、下位機種の命令コードを上位機種の命令コードに変換するコード変換テーブルが格納されている。

命令コードセクタ104は、内部データバス111上の命令コードか、コード変換メモリ105内の変換された命令コードかを選択し、実行ユニット102へ送出する。

メモリ108には、マイクロコンピュータ101がネイティブモードで実行するプログラム、エミュレーションモードで実行するプログラム、処理データが納められている。

間、レジスタ・メモリ間でのデータの転送、又は、レジスタへの直接データ転送、メモリへの直接データ転送を行う。ADDは、加算命令で、レジスタプラスレジスタ、レジスタプラスメモリ、メモリプラスレジスタ、レジスタプラスデータ、メモリプラスデータの処理を行い、計算結果をレジスタまたはメモリに格納する。SUBは、減算命令で、レジスタマイナスレジスタ、レジスタマイナスメモリ、メモリマイナスレジスタ、レジスタマイナスデータ、メモリマイナスデータの処理を行い、計算結果をレジスタまたは、メモリに格納する。IN A, nは、入出力命令で、第2オペランドで指定している入出力装置103のアドレスnに格納してあるデータをアキュムレータに転送する。OUT A, nは、入出力命令で、第2オペランドで指定している入出力装置103のアドレスnにアキュムレータの内容を転送する。INTは、割込命令で、INTが実行されるとPC115とPSW116をスタック領域に退避し、所定割込みベクタへ分岐する。そして、PSW116内

実行ユニット102内には、プログラム・カウンタ(以下PCと記す)115、プログラム・ステータス・ワード(以下PSWと記す)116、レジスタ群117が含まれている。

PSW116には、ネイティブモード、エミュレーションモードを設定するモード設定フリップフロップ118が設置されている。他にZフラグ、キャリーフラグ等も含まれているが本図中には図示していない。このモード設定フリップ・フロップ118が“1”の場合、シングルチップマイコン101はネイティブモードとなり、フェッチした命令コードを上位機種用の命令コードとして解釈、実行する。また、モード設定フリップフロップ118が“0”の場合、シングルチップマイコン101は、エミュレーションモードとなり、フェッチした命令コードを下位機種用の命令コードとして処理する。

以下に実行ユニット102が実行する命令処理を第2図のフローチャートを参照しながら説明する。MOVは、転送命令で、レジスタ・レジスタ

のモード設定フリップフロップ118を強制的にセットし、ネイティブモードとなる。RBT1命令は、主プログラムへの復帰命令で、この命令を実行すると、スタック領域に退避されている割込命令時のPC、PSWの情報をリストアし、再び主プログラムに戻る。

次に情報処理装置100の動作を説明する。この情報処理装置100はQUEUE 106を用いているため、実行ユニット102が命令コードの実行をしていても、フェッチユニット107の指示でメモリ108からのフェッチが行われ、QUEUE 106に書えらる。QRDY 信号109がアクティブレベル“1”であれば、QUEUE 106からは、命令コードが次々と送出されるため、前の命令コードが実行ユニット102に取り込まれると次の命令はすでにQUEUE 106から内部データバス111に送出されている。ここまでは、ネイティブ・モードにもエミュレーションモードにも共通の動作である。続く動作は、ネイティブモードとエミュレー

ションモードとに分けて説明する。

まず最初にネイティブモードの場合を説明する。内部データバス111上の命令コードは、上位機種用の命令コードである。実行ユニット102はネイティブモードで動作しているため、命令コードセレクト104は内部データバス111上の命令コードを選択し、実行ユニット102に出力する。実行ユニット102は、この命令コードをデコードして、第2図で説明した各命令を処理する。このときは、すでに次の命令コードが内部データバス111上に出力されており、前の命令コードの処理が終わるのを待っている。この様に第5図に示す通り、命令コードのフェッチ、命令コードの実行が平行して行われるため、第4図のQUEUEがない場合に比べ処理時間が短縮される。

次に、エミュレーションモードの場合を説明する。内部データバス111上の命令コードは、下位機種の命令コードである。下位機種の命令コードは、コード変換メモリ105に送られ、コード変換メモリ105内のコード変換テーブルによ

って上位機種の命令コードに変換される。すなわち、コード変換メモリ105は、F位機種用MOV命令の命令コードが入力すると、下位機種用MOV命令と同じ処理を行う上位機種用MOV命令の命令コードに変換し、同様に下位機種用ADD命令の命令コードが入力すると、下位機種用ADD命令と同じ処理を行う上位機種用ADD命令の命令コードに変換する。実行ユニット102がエミュレーションモードで動作しているため、命令コードセレクト104は、コード変換メモリ105の命令コードを選択し、上位機種命令に変換された命令コードを実行ユニット102に出力する。実行ユニット102はこの命令コードをデコードしてMOVであれば転送をADDであれば加算をSUBであれば、減算を実行する。命令コードを変換した後の実行ユニット102の動作は第2図で説明したものと同様である。そして、このときはすでに次の命令コードがコード変換されている。以上の様に、第7図に示す通り、命令コードのフェッチ、命令コードの変換、命令コードの実行が

平行して行われるため、第6図のQUEUEがない場合に比べ、処理時間が短縮される。

(発明の効果)

以上説明したように本発明は下位機種の命令セットを実行する機能と上位機種の命令セットを実行する機能を備えているため、下位機種用と上位機種用の両方のソフトウェアを実行できるという利点がある。

また、本発明に関する情報処理装置には、命令コード変換メモリを用いているため、2つのマイコンを搭載することなく上位機種の命令セットを実行する機能をもったマイコンだけで下位機種の命令も処理できるので、今までの下位機種用のマイコンと上位機種用のマイコンの2つを使っていた情報処理装置に比べ小型化でき、また、低コスト化も可能であるという利点もある。

さらに、本発明に関する情報処理装置には、QUEUEを用いているため、1命令の処理が終わるたびにメモリから命令コードを読み出すのではなく、命令コードのフェッチ、命令コードの実

行が平行して行われるため、プログラム処理速度が速いという利点がある。

4. 図面の簡単な説明

第1図は本発明における情報処理装置の構成図、第2図はネイティブモード時のフローチャート、第3図はエミュレーションモード時のフローチャート、第4図はQUEUEを用いないときのタイムチャート、第5図はQUEUEを用いたときのタイムチャート、第6図は本発明における情報処理装置においてQUEUEを用いない場合のタイムチャート、第7図は本発明における情報処理装置においてQUEUEを用いる場合のタイムチャート、第8図は従来のエミュレーション機能の付加されていない上位機種の構成図、第9図は従来の下位機種用エミュレーション機能付の上位機種の構成図である。

102……実行ユニット、103……入出力装置、104……命令コードセレクト、105……コード変換メモリ、106……QUEUE、107

……フロッピユニット、108……メモリ。
 109……QRDY信号、110……QFULL
 信号、111……内部データベース、112……外
 部データベース、113……内部アドレスバス、
 114……外部アドレスバス、115……PC、
 116……PSW、117……レジスタ、118
 ……モード設定フリップフロップ、801……
 CPU、802……メモリ、803……入出力終
 端、901……上位機専用マイコン、902……
 下位機専用マイコン、903……エミュレーシ
 ョン制御装置、904……メモリ、905……アド
 レスバス、906……データベース。

代理人 弁理士 内 原 啓

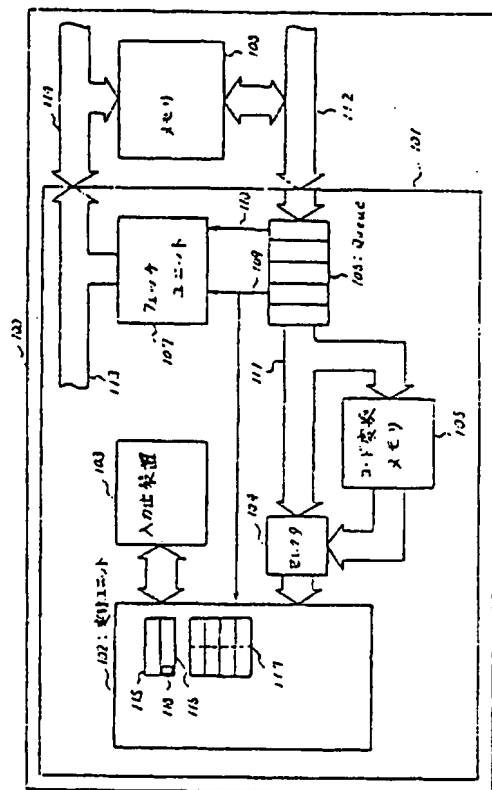
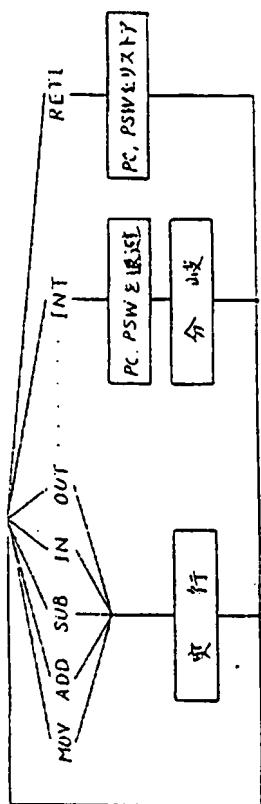


圖 1



第 2 圖

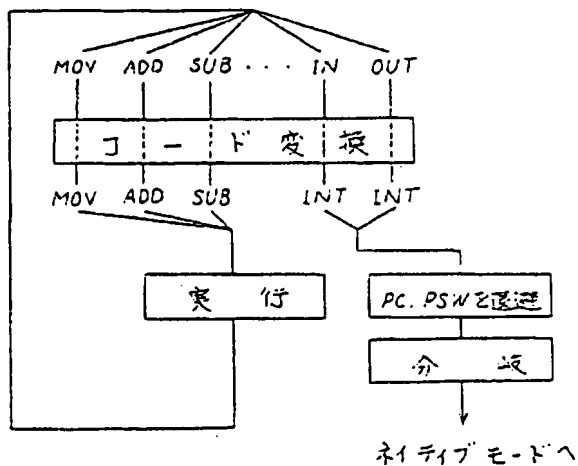
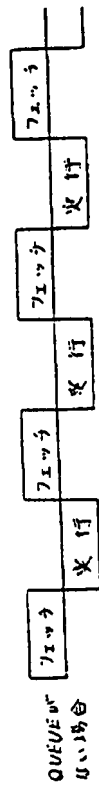
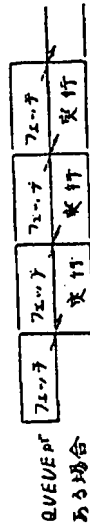


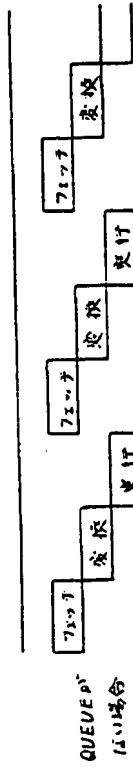
圖 3 梁



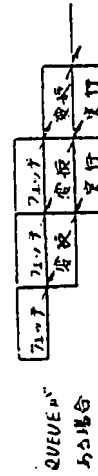
第 4 図



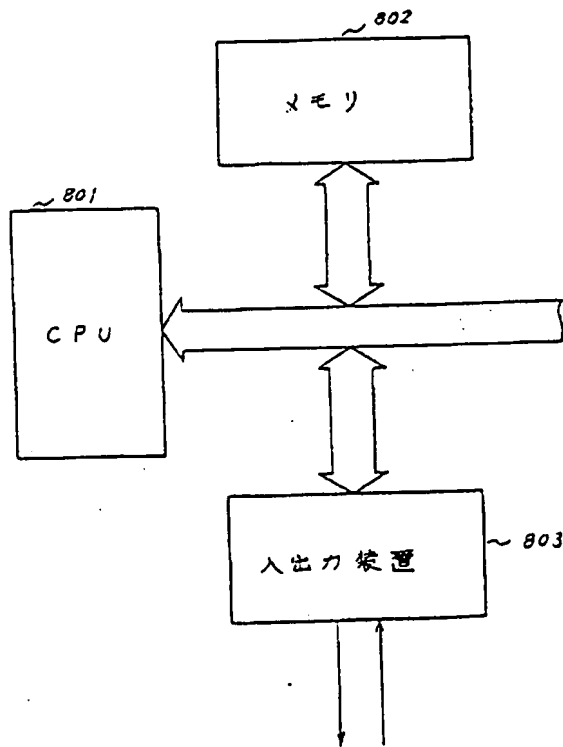
第 5 図



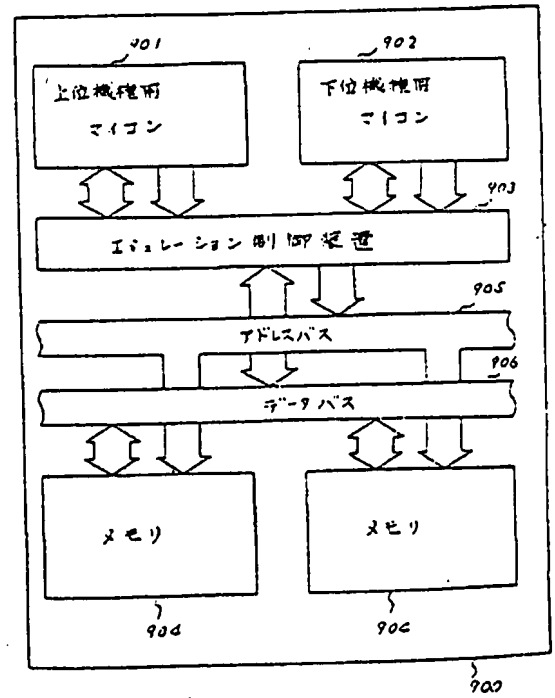
第 6 図



第 7 図



第 8 図 (従来例)



第 9 図 (従来例)